CS-308-2014 Final Report

TU - 09

Tomato Plucker

Revant Babu 100050054
Kumar Teja** 100050062
Sachin R 100050073
Gautam Sumu 100050058

Table of Contents

1. Introduction

We have built a tomato plucker bot. The motivation for our project is that nowadays people in cities people have started using greenhouses for growing vegetables etc. And we, through our project want to help automate the greenhouses more. Automating everything in the greenhouse is desirable to increase the popularity of the greenhouses and to decrease the amount of work people have to put in the greenhouses, the cost of labour in cities is very high. Our project helps in automating harvesting of vegetables. We have chosen tomatoes to harvest and our project detects ripe tomatoes in a prototype of the green house and plucks them.

2. Problem Statement

We are trying to automate the process of mapping and plucking ripe tomatoes in a particular green house setting. We are assuming user to be the owner of the greenhouse who wants to check for ripe tomatoes and pluck as many as he needs. Our project is trying to solve a problem in automating the greenhouse by doing automatic detection and plucking of ripe tomatoes.

We are constraining that the tomatoes are on the border of the trough - This constraint comes because it is tough to estimate the depth of the tomato away from the border just using one camera.It can be achieved using 2 cameras but space is a constraint.

We are also considering single tomatoes and not a bunch - Tomatoes come in bunches sometime. And the way we are planning to implement pluck is to cut the stem above the tomato, this wont be possible if we are considering tomato bunches, hence we are constraining ourselves to single tomatoes.

3. Requirements

3.1 Functional Requirements

- Monitoring Mode
  - Maps ripe tomatoes in 2D
  - Number of ripe tomatoes per row etc
- Collection Mode
  - Pluck tomato specified by row and column number
- Controllable remotely
  - Android/Laptop controllable using wireless communication

3.2 Non-Functional Requirements

- Adaptability
  - Should keep trying to cut a located tomato
  - Detect partially visible tomatoes
- Factor of safety
  - Should avoid collecting unripe tomatoes or damage tomato plants
- Response time
  - The response time should be reasonable
- Modularity and extensibility
  - The code written should be modular and extensible

3.3 Hardware Requirements

We will be using the following hardware interfaces for our project -

- Camera Sensors for detecting the tomatoes using image processing
- ZigBee module for wireless communication
- Gripper arm to pluck the tomatoes
- Servo Motors for moving the constructed Arm arrangement

3.4 Software Requirements

We will be capturing images using the camera sensor to detect the tomatoes. These images will be processed real-time on a computer running Windows. We will use image processing techniques for detection. For Image Processing we are using OpenCV/python.

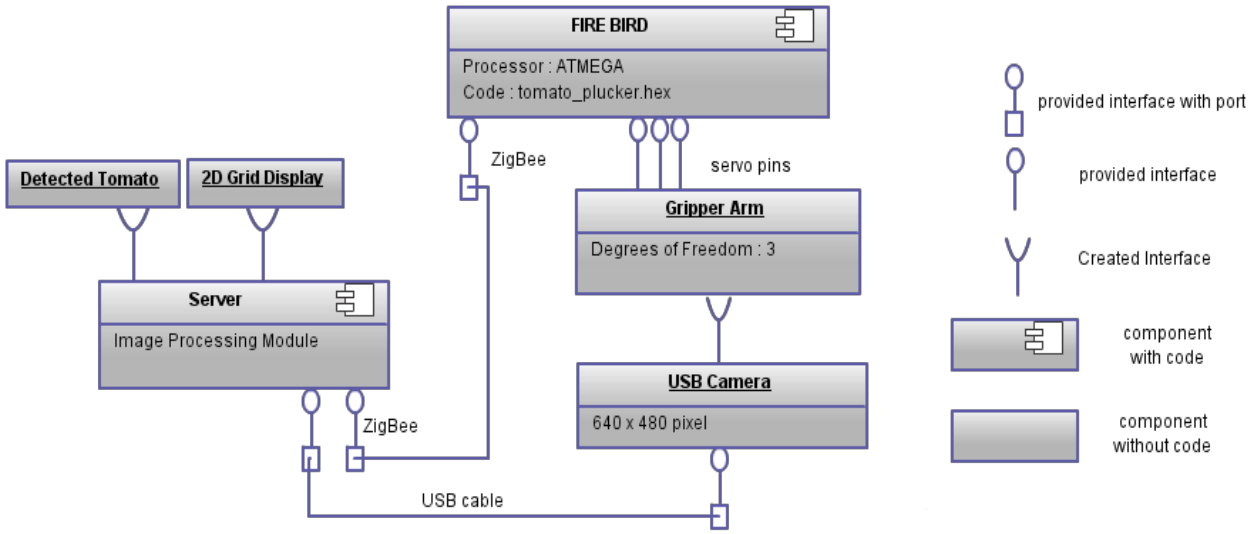
pygame library for generating 2D plot of tomatoes

zigBee serial communication module

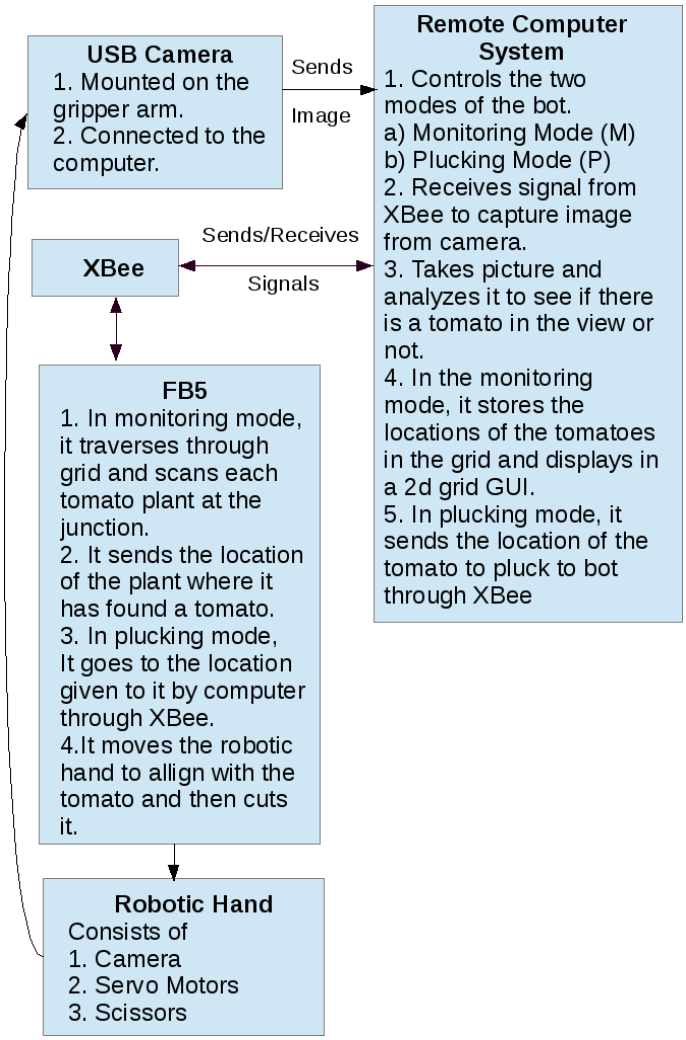
After detection we have to go and pluck tomatoes. We will use the robotic arm for plucking. So we might use any available code for moving the robotic arm.

4. System Design

System Architecture Design :



Flow Diagram :



5. Working of the System and Test results

Set Up :

- We've the field emulated by the rectangular grid with rows and columns
- Tomato plants are present on alternate columns at points of intersections
- We've mounted robotic arm on the bot with camera attached to it

Working:

- Initially bot is in idle state at start position, row:0, column:0
- It can be instructed to be in monitoring mode(give input 1) or plucking mode(give input 2) via remote laptop.

Monitoring Mode:

- In monitoring mode, the bot starts at row:0 and col:0 and moves around the field by line following.
- It stops at every node(points of intersection), by arrangement the tomato plant is on the right of it.
- Its arm is aligned to be in the plane of plant. The arm now moves from 120 degrees(top) and starts decrementing by 5 degrees
- At every 5 degrees, it takes a pic, checks if tomato is detected in the pic.
    - if detected it notes down the row and column number where it is found and moves on to the next node
    - if not found it continues decrementing till 0 degrees, which means no tomato found at this row, column
- Once the end of grid is reached, it send the information, no. of tomatoes found and the list of <row,col> where they are found.

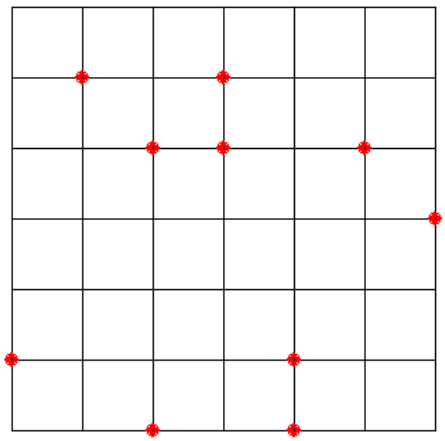


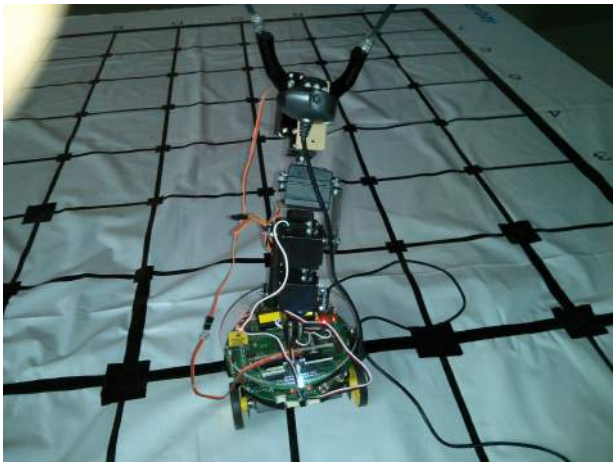
Fig : 2D map generated by monitoring

Plucking Mode:

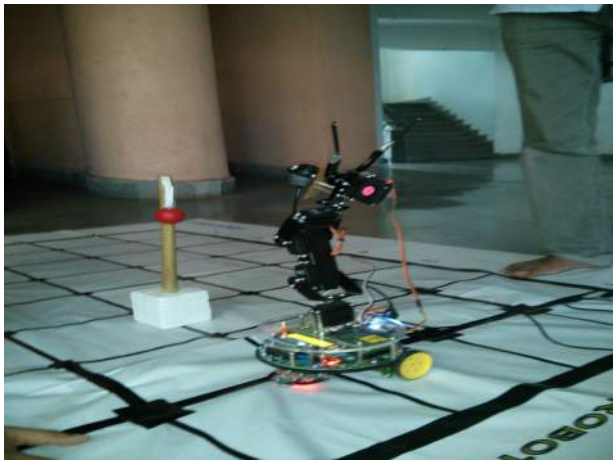
- In plucking mode system first prompts the user to enter row and column number from which tomato is to be plucked
- Then the bot moves to corresponding row and column by line following

- It tries to detect the fruit just like in monitoring mode
- Once tomato is detected, now the arm and bot are slightly adjusted until the detected tomato aligns to center of the camera
- If by chance, the tomato is not detected in first iteration (120 to 0 degrees) or it moves out of window while adjusting, the process is again repeated since we know tomato is present there.
- It repeats for maximum 5 times and sends fail message.
- If alignment is successful, by arrangement the cutters arranged are on top of fruit around the stem, so the cutters close at high speed and cut the stem
- In additionally pulls down in case , the stem was not cut.
- The cut fruit is collected at the basket below.

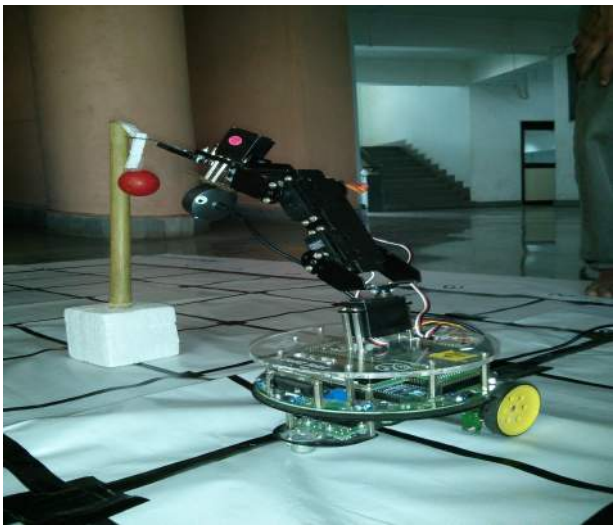
Pictures of working of the bot :



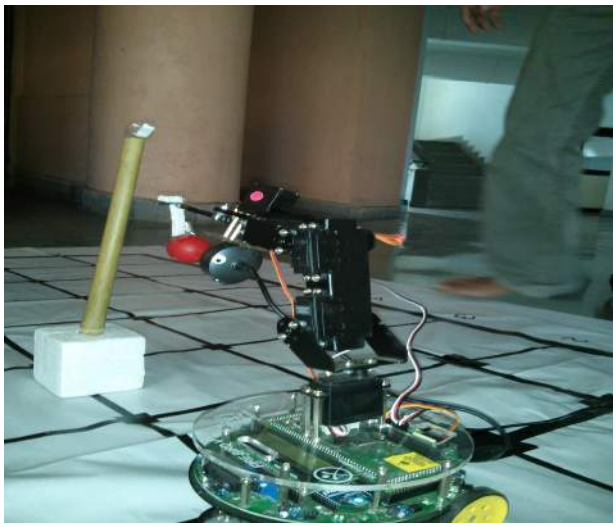
2D grid: A prototype of the field



The bot and the prototype of tomato plant



The bot detecting the tomato



Tomato being plucked

Functionality	Testing Strategy
Monitoring Mode	-Modeled the tomato field as rectangular grid. -Moved the bot around the grid to detect tomatoes -2D map generated contains most of the tomatoes
Detection Mode	-Is the bot moving to specified position -Is it able to detect and align the tomato to center -If it fails first time, is it retrying
Serial Communication	Values being transferred are printed both at server and on the bot using lcd screen

6. Discussion of System

Components of our project that worked as per plan -

1.Monitoring mode :

- worked as per plan, bot moves around the grid and generates 2D grid
- Some issues were faced while line following, missing one node would effect the entire calculations, moving exactly 90 deg right etc.

2.Plucking mode :

- Bot moves to the given location and aligns the tomato to center of camera and cuts the stem, it additionally plucks down if cutting misses
- The blades need to be sharp enough to cut, rest is working fine
- Same issues of line following as in monitoring mode

3.Remotely controllable :

- Initially thought of Android app, but it required bluetooth module which has to be fixed in Zigbee module slot
- We were already using the zigbee slot so we controlled remotely via the laptop using same zigbee module

Features not discussed in the SRS but added

- We added retrying, so retrying is when the bot misses to pluck a tomato it tries again and again because it knows that the tomato is there, it was not discussed in the SRS

Changes made in plan from SRS:

- The original plan was to use 2 cameras(cylinder model), so that it can detect tomatoes at any depth but given the size of cameras space was a constraint
- So we assumed that tomatoes are at border of the trough and continued with one

camera

7. Future Work

Reusable components -

- Image processing code of detecting tomato can be reused in projects that need to detect tomatoes or any other colored fruit
- Code for movement of Servo Motors according to the command by servo motors can be reused
- Modified gripper arm which can see using camera and cuts tomato using scissors
- Line follower code which creates a 2d map of grid by counting the number of rows and columns by analyzing the junction point
- Code to align the bot and the gripper arm to the center of the tomato

Possible Extensions -

- We can employ machine learning techniques in the image processing module to detect tomatoes in a much accurate manner
- Optimized plucking in plucking mode, collect specified number of tomatoes with minimum movement using the 2D map generated.
- Case of plucking a bunch of tomatoes that are located in the same place is not handled, it can be handled in the future
- Add an android network interface so that a person working in his office can monitor and pluck tomatoes

8. Conclusions

So in this project we were able to achieve our goal of mapping ripe tomatoes and plucking them. Though the system is not very robust but it is a good starting point to making of a very robust system.We could incorporate the points mentioned in future directions to make it more useful.

9. References

- Opencv Python Tutorial - http://docs.opencv.org/trunk/doc/py_tutorials/py_tutorials.html
- AVR studio basics : http://www.atmel.in/Images/novice.pdf
- AVRExperiments and manual https://github.com/akshar100/eyantra-firebird-resources/tree/master/Fire%20Bird%20V%20ATMEGA2560%20Robots%202010-12-29